Use of jQuery on HTML CSS

8

LEARNING OBJECTIVES

After completing this chapter, you will be able to understand:

- How to add jQuery to HTML pages.
- Various jQuery functions.
- Development with jQuery.
- How to use any text editor to code jQuery.

8.1 | Getting Started with jQuery

For jQuery you do not really need to set up any environment. As is the case with HTML, you can use any text editor you like. You can use the same editor you have used for coding HTML and CSS as well.

The only thing you need to make sure is that you add jQuery library to the page. There are two ways of doing this: First way is to add it from external sources like CDN or Google CDN or jQuery's own website or your own server. It really does not matter. The second way is to download jQuery library and keep it in the same place as "*home.html*". If you change the place, you just need to give the proper path of that location. For example, if some developer likes to keep this file in a folder named "*js*", your path will look like "*js/jquerylibrary.js*". In this case, *jquerylibrary.js* is the name of the current version of the library. The name can be anything you like as you are keeping it on your server. However, if you are using a hosted version from CDN then you cannot change the filename. Let us see the following example of using a hosted version of jQuery file.

```
<script src=“https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js”></script>
```

This is how you get jQuery file from a CDN. In our case, we have used Google CDN. As you already learned, this <*script*> tag will go under <*head*> tag.

Now, let us use our home page and start adding jQuery code to it.

8.2 | Home Page with jQuery

To start with let us add functionality to the search button we have added on the page. This button is intended to provide the list of all the products in the system which match the search criteria given by user in the search textbox.

In order to access the button in jQuery code, we need to add an "*id*" attribute to the <*button*> tag so that we can access it. Let us add "*id*" attribute first:

```
<button id=“searchbtn”>Search</button>
```

Now, we need to add jQuery code to access this button and capture the click action. So, when a user clicks on it, we can execute the code we want. Now, let us start building the jQuery code for the same. We will add another *<script></script>* block to add our jQuery code.

As you have learned in Chapter 5, we can place our code inside *$(document).ready(function() { }* to make sure we have all the HTML elements loaded properly before accessing them.

After doing this, our *<head>* section will look as follows:

```
<head>
       <script src=“https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js” >
</script>
       <script>
          $( document ).ready(function() {

          });
       </script>
       <title>MyEShop</title>
</head>
```

If you are not sure about the jQuery syntax we have used, please visit Chapter 5 again to clear your doubts on jQuery. Now, let us write the code to access button with *id* selector.

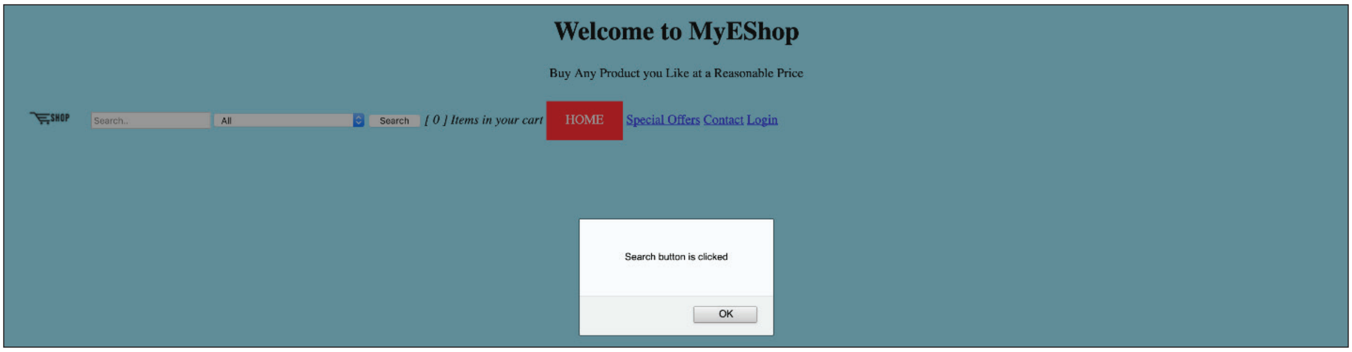
```
<script>
          $( document ).ready(function() {
             $( “#searchbtn” ).click(function() {

             });
          });
</script>
```

As you have learned, id selection is captured by using “*#*” and then the text mentioned in the “*id*” attribute. In our search button case, the id of the search button is “*searchbtn*” and hence we have used “*#searchbtn*” to select this particular button. Then we need to capture the click event so we have added “*.click()*” to capture this event. However, we need to place our code in between that block so our code will go inside “*$(“#searchbtn”).click(**function**() { });*”. Let us move on to add an alert box to make sure the click event is properly capturing the event.

This is how our code looks like for the alert box.

```
<script>
          $( document ).ready(function() {
             $( “#searchbtn” ).click(function() {
                alert(“Search button is clicked”);
             });
          });
</script>
```

Now, let us save an and open this file in a browser window; we will see the following result.



Welcome to MyEShop

Buy Any Product you Like at a Reasonable Price

[0] Items in your cart

Search button is clicked

Once the page is loaded, a pop-up box is visible which shows a message. This appears when we clicked on the “*Search*” button. Now we know our click event is working.

In order to make a proper search and display function, we need to consider the text from the search text box and text from the category dropdown. We will use these two values to filter the result from the database to get the selected products back. So, let us code to capture these two values.

First read the textbox value. For this, we have to do the same thing we did for the button. We need to add the “*id*” attribute to the textbox so we can use *id* selector in jQuery to get access to this particular textbox.

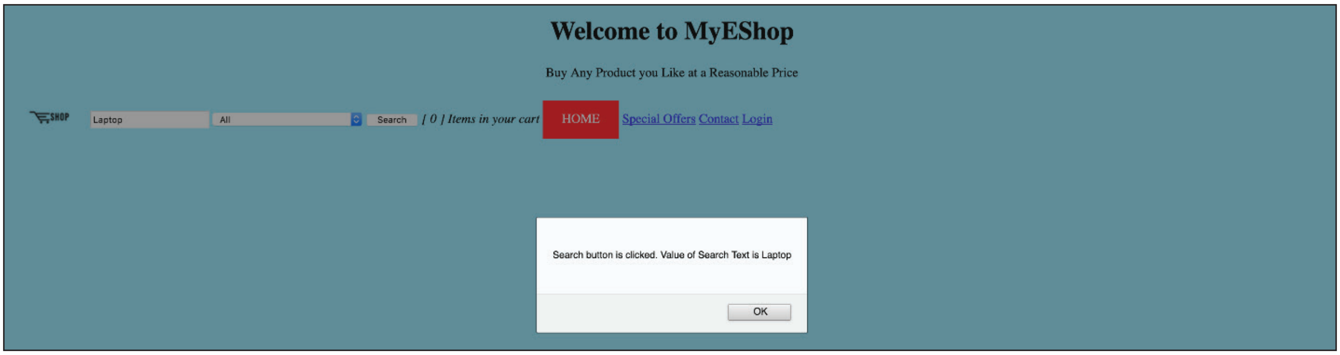
This is how our code looks like after adding the id attribute.

```
<input id=“searchtxt” type=“text” placeholder=“Search..”>
```

Now let us add code on the jQuery side to access this and capture the value from it. In order to get the value from this textbox, we need to use “*.val()*” function. This is how our code will look after adding it.

```
<script>
          $( document ).ready(function() {
             $( “#searchbtn” ).click(function() {
                alert(“Search button is clicked. Value of Search Text is” +
$(“#searchtxt”).val());
             });
          });
</script>
```

For now, we are just adding it in the alert box so we can see the captured value in the popup. Now let us run this in a browser window.



Welcome to MyEShop

Buy Any Product you Like at a Reasonable Price

[0] Items in your cart

Search button is clicked. Value of Search Text is Laptop

As you can see in the screenshot, we can get the captured value from the textbox. Since we can also filter from the category item, we need to capture the selected category as well. For this we will repeat the id selector procedure.

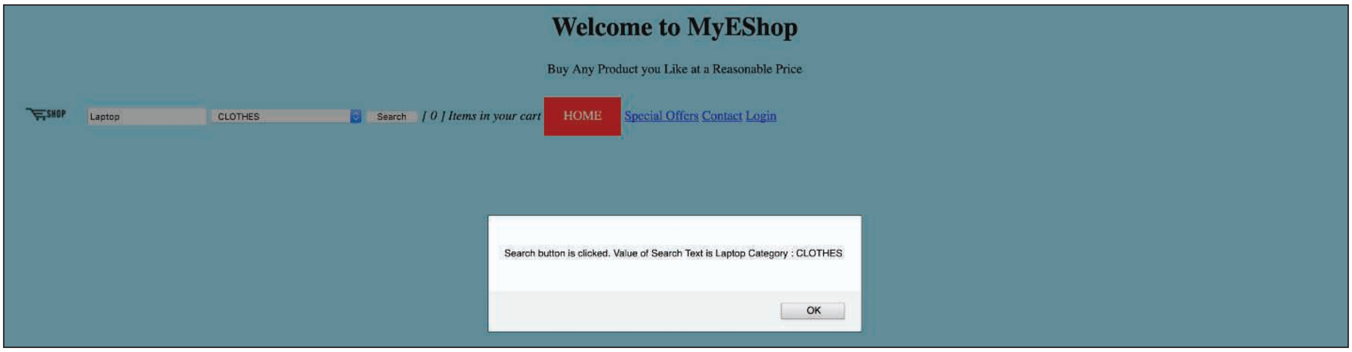
This is what our code looks like when we add id to the select element.

```
<select id=“categorysel”>
       <option>All</option>
       <option>CLOTHES </option>
       <option>FOOD AND BEVERAGES </option>
       <option>HEALTH & BEAUTY </option>
       <option>SPORTS & LEISURE </option>
       <option>BOOKS & ENTERTAINMENTS </option>
</select>
```

Now, let us use *id* selector on jQuery side to access this select element. To capture selected option, we can use the following code. Please note that here we are not using “*.val()*” but using “*.html()*”.“*.html()*” is used to get data from the element as elements do not have value but only content data. On the other hand, input box has value.

```
<script>
          $( document ).ready(function() {
             $( “#searchbtn” ).click(function() {
                alert(“Search button is clicked. Value of Search Text is” +
$(“#searchtxt”).val() + “ Category : ” + $(“#categorysel”).children(“:selected”).
html());
             });
          });
</script>
```

Also, note that to get a value of selected option, we are using “*(“:selected”)*” on all the children of the category dropdown. jQuery traverses all the children and identifies the selected one. After that, as we just discussed, we get “*html()*” to fetch value from the element. See the following screenshot:



Welcome to MyEShop

Buy Any Product you Like at a Reasonable Price

Search button is clicked. Value of Search Text is Laptop Category : CLOTHES

This is good start for you to continue your journey into developing this application further. You can add any action you like in place of alert statement. However, to make it functional we will be calling web services from this code block and parsing the result we are going to get from the webserver. We will work on this ajax call code in Chapter 24, once our web service is ready in Chapters 22 and 23.

In the end, this is how our complete file looks.

```
<!DOCTYPE html>
<html>
    <head>
        <script src=“https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js”
></script>
        <script>
            $( document ).ready(function() {
                $( “#searchbtn” ).click(function() {
                    alert(“Search button is clicked. Value of Search Text is ” +
$(“#searchtxt”).val() + “ Category : ” + $(“#categorysel”).children(“:selected”).
html());
                });

            });
        </script>
        <title>MyEShop</title>
    </head>
    <body style=“background-color:#A2E9FF;”>
        <h1 style=“text-align:center;”>Welcome to MyEShop</h1>
        <p style=“text-align:center;”>Buy Any Product you Like at a Reasonable Price</p>

        <table>
            <tr>
                <td><img src=“myeshoplogo.png” alt=“myeshop logo” width=“50px”
height=“60px”></td>
                <td><form> <input id=“searchtxt” type=“text” placeholder=“Search..”>
</form></td>
                <td>
                    <select id=“categorysel”>
                        <option>All</option>
                        <option>CLOTHES </option>
                        <option>FOOD AND BEVERAGES </option>
                        <option>HEALTH & BEAUTY </option>
                        <option>SPORTS & LEISURE </option>
                        <option>BOOKS & ENTERTAINMENTS </option>
                    </select>
                </td>
                <td><button id=“searchbtn”>Search</button></td>
                <td><span ><i> [ 0 ] Items in your cart </i> </span></td>
                <td><a style=“background-color: red;color: white;padding: 1em 1.5em;-
text-decoration: none;text-transform: uppercase;” href=“home.html”>Home</a></td>
                <td><a href=“specialofferrs.html”>Special Offers</a></td>
                <td><a href=“contact.html”>Contact</a></td>
                <td><a href=“login.html”>Login</a></td>
            </tr>
        </table>

    </body>
</html>
```

QUICK CHALLENGE

Tom was able to design an HTML page with CSS and add jQuery code to change page content based on selected value from a dropdown. Help him write the code to add ajax calls to fetch the latest product reviews from Google or other websites. Also, he was thinking of adding a box to show all the Facebook posts related to the product from his shop’s Facebook page. Help him figure out the Facebook integration.

Summary

As you have seen, it is very easy to add jQuery to any page and use it without installing anything on the server or client side. The only thing we need to do is get the jQuery library. jQuery code can be written in any simple text editor as well. So, getting started with jQuery is a simple process.

In this chapter, we have learned the following concepts:

1. Process for accessing jQuery library.
2. Adding jQuery library to HTML pages.
3. Various jQuery functions that we can use on the page.
4. Using id selector to get a particular element to work on.

In Chapter 9, we will use Bootstrap to add some responsive behavior to our pages, so that they render properly in any type of device with varying screen sizes.

Multiple Choices Questions

1. Which of the following options is used to alternately expand and collapse a page element?
   - **(a)** `.toggle()`
   - **(b)** `.trigger()`
   - **(c)** `.stopPropagation()`
   - **(d)** `.hover()`
2. Which of the following options is utilized to return/set arbitrary data to/from an element?
   - **(a)** `.item()`
   - **(b)** `.data()`
   - **(c)** `.all`
   - **(d)** None of the above
3. JQuery is a W3C standard.
   - **(a)** True
   - **(b)** False
4. Which of the following JQuery methods is used to hide selected elements?
   - **(a)** `hide()`
   - **(b)** `display(none)`
   - **(c)** `hidden()`
   - **(d)** `visible(false)`
5. Which scripting language is JQuery written in?
   - **(a)** C++
   - **(b)** JavaScript
   - **(c)** C#
   - **(d)** VBScript

Review Questions

1. How do you get the selected values from a dropdown?
2. What are the advantages of using jQuery from a CDN server over using a local downloaded copy on the server?
3. What are the disadvantages of using jQuery from a CDN server over using a local downloaded copy on the server?
4. How do we capture a click event for a button?
5. How do we use an alert popup to notify user?

Exercise

Carry on with the development of this page. Think of programming logic you can add on this page to improve this further. Also add jQuery on the new pages you have created in the exercise in Chapter 7.

Project Idea

Take the example of the school management system from Chapter 7's Project Idea section. Add interactivity to the HTML and CSS pages you have created for the requirements mentioned in that section. Make sure you add validation to the new student form.

Recommended Readings

1. David Sawyer McFarland. 2014. *JavaScript & jQuery: The Missing Manual*. O'Reilly: Massachusetts
2. John Pollock. 2014. *jQuery: A Beginner's Guide*. McGraw-Hill Education: New York
3. Jon Duckett. 2014. *Web Design with HTML, CSS, JavaScript and jQuery Set*. Wiley: New Delhi